New Lower Bounds for Halfspace Emptiness*

Jeff Erickson

Computer Science Division
University of California
Berkeley, CA 94720–1776
jeffe@cs.berkeley.edu
http://http.cs.berkeley.edu/~jeffe

Abstract

We derive a lower bound of $\Omega(n^{4/3})$ for the halfspace emptiness problem: Given a set of n points and n hyperplanes in $\mathbb{R}^5$, is every point above every hyperplane? This matches the best known upper bound to within polylogarithmic factors, and improves the previous best lower bound of $\Omega(n \log n)$. The lower bound applies to partitioning algorithms in which every query region is a polyhedron with a constant number of facets.

1. Introduction

The halfspace emptiness problem asks, given a set of points and a set of halfspaces, whether any halfspace contains a point. In this paper, we derive new lower bounds for the time required to solve this problem, generalizing earlier lower bounds for Hopcroft's point-line incidence problem [16].

In this paper, we will consider the following formulation of the problem: Given a set of points and hyperplanes, is every point above every hyperplane? Using linear programming [13, 21, 24, 25], we can decide in linear time whether the union of a set of halfspaces is $\mathbb{R}^d$. If it is, then *every* input point must lie in a halfspace. Otherwise, by aplying an appropriate projective transformation, which we can also find in linear time, we can ensure that the halfspaces miss the point $(0, 0, \ldots, 0, \infty)$. If we use the duality transformation $(a_1, a_2, \ldots, a_d) \longleftrightarrow \sum_{i=1}^{d-1} a_i x_i = a_d + x_d$, then a point p is above a hyperplane h if and only if the dual point h^* is above the dual hyperplane p^*. Thus, in this formulation, the halfspace emptiness problem is self-dual.

The best known algorithms for this problem were developed for its online version: Given a set of n points, preprocess it to answer halfspace emptiness (or reporting) queries. In two and three dimensions, we can easily build a linear-size data structure in $O(n \log n)$ time, that allows halfspace emptiness queries to be answered in logarithmic time [3, 10, 14]. In higher dimensions, a randomized algorithm due to Clarkson [12] answers halfspace emptiness queries in time $O(\log n)$ after $O(n^{\lfloor d/2 \rfloor + \varepsilon})$[1] preprocessing time. Matoušek [19] describes two halfspace emptiness data structures, one answering queries in time $O(n^{1-1/\lfloor d/2 \rfloor} \operatorname{polylog} n)$ time after $O(n \log n)$ preprocessing time, and the other answering queries in time $O(n^{1-1/\lfloor d/2 \rfloor} 2^{O(\log^* n)})$[2] after $O(n^{1+\varepsilon})$ preprocessing time. Combining Clarkson's and Matoušek's data structures, for a fixed parameter $n \le s \le n^{\lfloor d/2 \rfloor}$, one can answer queries in time $O((n \log n)/s^{1/\lfloor d/2 \rfloor})$ after $O(s \operatorname{polylog} n)$ preprocessing time [19, 1, 7]. For extensions and applications of halfspace range reporting, see [1, 2, 6, 7, 22, 20].

Given n points and m halfspaces, we can solve the offline halfspace emptiness problem in time

$$O\left(n \log m + (nm)^{\lfloor d/2 \rfloor / (\lfloor d/2 \rfloor + 1)} \operatorname{polylog}(n+m) + m \log n\right),$$

using either Clarkson's data structure or Matoušek's combined data structure, depending on the relative growth rates of n and m. In two and three dimensions, the time bound simplifies to $O(n \log m + m \log n)$. If $n > m$, we actually solve the problem in the dual, by building a data structure to report if any halfspace contains a query point.

*This research was partially supported by a GAANN fellowship. The author's current address is Computer Science Department, Duke University, Box 90129, Durham, NC 27708–0129.

[1] In bounds of this form, ε is an arbitrarily small positive constant. Multiplicative constants hidden by the big-Oh notation tend to infinity as ε approaches zero.

[2] The iterated logarithm $\log^* n$ is defined as $1 + \log^*(\lg n)$ when $n \ge 2$ and 1 when $n < 2$.

The only lower bound previously known for this problem is $\Omega(n \log m + m \log n)$, in the algebraic decision tree or algebraic computation tree models, by reduction from set intersection [26, 4]. Thus, the two- and three-dimensional algorithms are optimal, but there is still a large gap in dimensions four and higher.

In this paper, we derive a lower bound of $\Omega(n \log m + n^{2/3}m^{2/3} + m \log n)$ on the complexity of the halfspace emptiness problem in $\mathbb{R}^5$, matching known upper bounds up to polylogarithmic factors. We obtain marginally larger bounds in dimensions 9 and higher. Our lower bounds apply to *polyhedral partitioning algorithms*, a restriction of the class of partitioning algorithms introduced in [16]. Informally, a polyhedral partitioning algorithm covers space with a constant number of constant-complexity polyhedra, determines which points and halfspaces intersect each polyhedron, and recursively solves the resulting subproblems.

The basic approach is the same as the one used to prove lower bounds for Hopcroft's problem [16]. We first define *polyhedral covers*, and develop lower bounds on their combinatorial complexity. Our main result (Theorem 5.2) states that the running time of a polyhedral partitioning algorithm is bounded below by the polyhedral cover size of its input. The $\Omega(n^{4/3})$ lower bound then follows from the construction of a set of points and hyperplanes in $\mathbb{R}^5$, with all the points above all the hyperplanes, whose every polyhedral cover is that large. Our techniques also imply slightly better lower bounds for Hopcroft's problem in higher dimensions.

The paper is organized as follows. In Section 2, we define projective polyhedra and describe some of their basic properties. In Section 3, we state an important technical lemma concerning the separation of points and hyperplanes by polyhedra; the proof is given in the Appendix. We develop bounds on the size of polyhedral covers in Section 4. In Section 5, we formally define polyhedral partitioning algorithms and prove our main results. Finally, in Section 6, we offer our conclusions and suggest directions of further research.

2. Projective Polyhedra

Our lower bound argument relies heavily on certain properties of convex polytopes and polyhedra. Many of these properties are more easily proved, and have fewer special cases, if we state and prove them in projective space rather than affine Euclidean space. In particular, developing these properties in projective space allows us to more easily deal with unbounded and degenerate polyhedra and duality transformations. Everything we describe in this section can be formalized algebraically in the language of polyhedral cones and linear subspaces one dimension higher; we will give a much less formal, purely geometric treatment. For more technical details, we refer the reader to Chapters 1 and 2 of Ziegler's lecture notes [28].

The projective space $\mathbb{RP}^d$ can be defined as the set of lines through the origin in $\mathbb{R}^{d+1}$. Every k-dimensional linear subspace of $\mathbb{R}^{d+1}$ induces a $(k-1)$-dimensional *flat* f in $\mathbb{RP}^d$, and its orthogonal complement induces the *dual flat* f^*.

A *projective polyhedron* is a single closed cell, not necessarily of full dimension, in the arrangement of a finite number of hyperplanes in $\mathbb{RP}^d$. A *projective polytope* is a simply-connected polyhedron, or equivalently, a polyhedron that is disjoint from some hyperplane. Every projective polyhedron is (the closure of) the image of a convex polyhedron under some projective transformation, and every projective polytope is the projective image of a convex polytope. Every flat is also a projective polyhedron.

The *projective span* (or projective hull) of any subset $X \subseteq \mathbb{RP}^d$, denoted $\operatorname{span}(X)$, is the projective subspace of minimal dimension that contains it. The *relative interior* of a projective polyhedron is its interior in the subspace topology of its projective hull. A hyperplane *supports* a polyhedron if it intersects the polyhedron but not its relative interior. A flat has no supporting hyperplanes.

A *proper face* of a polyhedron is the intersection of the polyhedron and one or more supporting hyperplanes. Every proper face of a polyhedron is a lower-dimensional polyhedron. A *face* of a polyhedron is either a proper face or the entire polyhedron. We write $\Phi \leq \Pi$ to denote that a polyhedron Φ is a face of another polyhedron Π. The *dimension* of a face is the dimension of its projective hull. The dimension of the empty set is taken to be -1. The faces of a polyhedron form a lattice under inclusion. Every projective polyhedron has a face lattice isomorphic to that of a convex polytope, possibly of lower dimension.

The *apex* of a polyhedron Π, denoted $\operatorname{apex}(\Pi)$, is the intersection of all the supporting hyperplanes, or equivalently, the unique face of minimum dimension. The apex is empty if and only if the polyhedron is a polytope but not a single point; the apex is the whole polyhedron if and only if the polyhedron is a flat.

The *dual* or *polar* of a polyhedron Π, denoted Π^*, is the set of points whose dual hyperplanes intersect Π in one of its faces:

$$\Pi^* \triangleq \{p \mid (p^* \cap \Pi) \leq \Pi\}.$$

In other words, $p \in \Pi^*$ if and only if p^* either contains Π, supports Π, or completely misses Π. This definition generalizes both the polar of a convex polytope and the projective dual of a flat. We easily verify that Π^* is a projective polyhedron whose face lattice is the inverse of the face lattice of Π. In particular, Π and Π^* have the same number of faces. See [28, pp. 59–64] and [27, pp. 143–150] for similar definitions.

3. Polyhedral Separation

Let P be a set of points, let H be a set of hyperplanes, and let Π be a projective polyhedron in $\mathbb{RP}^d$. We say that Π *separates* P and H if Π contains P and the dual polyhedron Π^* contains the dual points H^*; that is, any hyperplane in H either contains Π, supports Π, or misses Π entirely. Both P and H may intersect the relative boundary of Π. We say that P and H are *r-separable* if there is a projective polyhedron with at most r faces that separates them.

The lower bound proofs in [16] relied on the following trivial observation: if we perturb a set of points and hyperplanes just enough to remove any point-hyperplane incidences, and every point is above every hyperplane in the perturbed configuration, then no point was below a hyperplane in the original configuration. The following technical lemma establishes the corresponding, but no longer trivial, property of r-separable configurations. Informally, if a configuration is not r-separable, then arbitrarily small perturbations cannot make it r-separable.

Technical Lemma 3.1. *Let H be a set of hyperplanes in $\mathbb{RP}^d$. For all r, the set of point configurations $P \in (\mathbb{RP}^d)^n$ such that P and H are r-separable is topologically closed.*

For a proof of this lemma, see the appendix.

4. Incidences and Polyhedral Covers

A *r-polyhedral cover* of a set P of points and a set H of hyperplanes is an indexed set of subset pairs $\{(P_i, H_i)\}$, where $P_i \subseteq P$ and $H_i \subseteq H$, such that

1. For each index i, P_i and H_i are r-separable.
2. For each point $p \in P$ and hyperplane $h \in H$, there is some index i such that $p \in P_i$ and $h \in H_i$.

We emphasize that the subsets P_i are not necessarily disjoint, nor are the subsets H_i. We refer to each subset pair (P_i, H_i) in a polyhedral cover as a *r-polyhedral minor*. The *size* of a cover is the sum of the sizes of the subsets P_i and H_i.

Let $\pi_r(P, H)$ denote the size of the smallest r-polyhedral cover of P and H. Let $\pi^\circ_{d,r}(n, m)$ denote the maximum of $\pi_r(P, H)$ over all sets P of n points and H of m hyperplanes in $\mathbb{RP}^d$ such that no point lies on any hyperplane. When the subscript r is omitted, we take it to be a fixed constant.

Finally, let $I(P, H)$ denote the number of point-hyperplane incidences between P and H.

Lemma 4.1. *Let P be a set of n points and H a set of m hyperplanes, such that no subset of s hyperplanes contains t points in its intersection. If P and H are r-separable, then $I(P, H) \le r(s + t)(n + m)$.*

Proof: Let Π be a polyhedron with r faces that separates P and H. For any point $p \in P$ and hyperplane $h \in H$ such that p lies on h, there is some face f of Π that contains p and is contained in h. For each face f of Π, let P_f denote the points in P that are contained in f, and let H_f denote the hyperplanes in H that contain f.

Since no set of s hyperplanes can all contain the same t points, it follows that for all f, either $|P_f| < t$ or $|H_f| < s$. Thus, we can bound $I(P, H)$ as follows.

$$\begin{aligned} I(P,H) &\le \sum_{f \le \Pi} I(P_f, H_f) \\ &= \sum_{f \le \Pi} (|P_f| \cdot |H_f|) \\ &\le (s+t) \sum_{f \le \Pi} (|P_f| + |H_f|) \end{aligned}$$

Since Π has r faces, the last sum counts each point and hyperplane at most r times. $\square$

The next lemma shows that sufficiently small perturbations of a point-hyperplane configuration cannot decrease its polyhedral cover size.

Lemma 4.2. *Let P be a set of n points and H a set of m hyperplanes in $\mathbb{RP}^d$. For all $Q \in (\mathbb{RP}^d)^n$ sufficiently close to P, $\pi_r(Q, H) \ge \pi_r(P, H)$.*

Proof: Let P' be a subset of P, and for any other set Q of n points, let Q' be the corresponding subset of Q. Let H' be a subset of H. Technical Lemma 3.1 implies that there is an open set $\mathcal{U}(P', H') \subseteq (\mathbb{RP}^d)^n$ such that for any $Q \in \mathcal{U}(P, H)$, if Q' and H' are r-separable, then P' and H' are also r-separable.

Let $\mathcal{U}$ be the intersection of these $2^n 2^m$ open sets:

$$\mathcal{U} = \bigcap_{P' \subseteq P} \bigcap_{H' \subseteq H} \mathcal{U}(P', H').$$

For all $Q \in \mathcal{U}$, every r-polyhedral minor of Q and H corresponds to a r-polyhedral minor of P and H. Thus, for any r-polyhedral cover of Q and H, there is a corresponding r-polyhedral cover of P and H with exactly the same size. $\square$

Theorem 4.3. $\pi_2^\circ(m,n) = \Omega(n + n^{2/3}m^{2/3} + m)$.

Proof: Let P be a set of n points and H a set of m lines in the plane such that $I(P,H) = \Omega(n+n^{2/3}m^{2/3}+m)$. Such a point-line configuration was first constructed by Erdős; see, for example, [15, p. 112].

Consider any subsets $P_i \subseteq P$ and $H_i \subseteq H$ such that P_i and H_i are r-separable. Since two distinct lines in the plane intersect in at most one point, Lemma 4.1 implies that $I(P_i, H_i) \le 4r(|P_i| + |H_i|)$. It follows that any collection of r-polyhedral minors that includes every incidence between P and H must have size at least $I(P,H)/4r$. Thus, $\pi_{2,r}(P,H) = \Omega(n+n^{2/3}m^{2/3}+m)$.

Finally, Lemma 4.2 implies that we can perturb P slightly, removing all the incidences, without decreasing the polyhedral cover size. $\square$

Our higher-dimensional lower bounds are based on the following generalization of the Erdős configuration, originally described in [16].

Lemma 4.4. *For any $\lfloor n^{1/d} \rfloor < m$, there exists a set P of n points and a set H of m hyperplanes in $\mathbb{R}^d$, such that $I(P,H) = \Omega(n^{1-2/d(d+1)}m^{2/(d+1)})$ and any d hyperplanes in H intersect in at most one point.*

Theorem 4.5. $\pi_d^\circ(n,m) =$
$\Omega\left(\sum_{i=1}^{d}(n^{1-2/i(i+1)}m^{2/(i+1)} + n^{2/(i+1)}m^{1-2/i(i+1)})\right)$

Since our d-dimensional lower bound only improves our $(d-1)$-dimensional lower bound for certain values of n and m, we have combined the lower bounds from all dimensions $1 \le i \le d$ into a single expression. If the relative growth rates of n and m are fixed, the entire sum can be reduced to a single term.

Following the terminology in [16], we say that a point-hyperplane configuration in $\mathbb{R}^d$ is *monochromatic* if every point lies above every hyperplane. Let $\hat{\pi}_{d,r}(m,n)$ denote the maximum of $\pi_r(P,H)$ over all monochromatic configurations of n points and of m hyperplanes in $\mathbb{R}^d \subset \mathbb{RP}^d$.

We define the family of functions $\sigma_d : \mathbb{R}^{d+1} \to \mathbb{R}^{\binom{d+2}{2}}$ as follows.

$$\sigma_d(x_0, x_1, \ldots, x_d) \triangleq (x_0^2, x_1^2, \ldots, x_d^2, \sqrt{2}\,x_0x_1, \sqrt{2}\,x_0x_2, \ldots, \sqrt{2}\,x_{d-1}x_d)$$

For any two vectors $u, v \in \mathbb{R}^{d+1}$, we have $\langle \sigma_d(u), \sigma_d(v)\rangle = \langle u, v\rangle^2$, where $\langle\cdot,\cdot\rangle$ is the standard vector inner product. In a more geometric setting, σ_d maps points and hyperplanes in $\mathbb{R}^d$, represented in homogeneous coordinates, to points and hyperplanes in $\mathbb{R}^D$, also in homogeneous coordinates, where $D = \binom{d+2}{2} - 1 = d(d+3)/2$. If the point p is incident to the hyperplane h, then $\sigma_d(p)$ is also incident to $\sigma_d(h)$; otherwise, $\sigma_d(p)$ is above $\sigma_d(h)$.

Applying σ_2 to the Erdős point-line configuration, or σ_d to its d-dimensional generalization, gives us a configuration of points and hyperplanes with several incidences, with no point below any hyperplane, and every d hyperplanes intersecting in at most a single point. The arguments in Theorem 4.3 immediately imply the following lower bounds.

Theorem 4.6. $\hat{\pi}_5(n,m) = \Omega(n + n^{2/3}m^{2/3} + m)$.

Theorem 4.7. *For all $D \ge d(d+3)/2$, $\hat{\pi}_D(n,m) = \Omega\left(\sum_{i=1}^{D}(n^{1-2/i(i+1)}m^{2/(i+1)} + n^{2/(i+1)}m^{1-2/i(i+1)})\right)$*

It is clear that $\hat{\pi}_3(n,m) = \Theta(n+m)$, since both the convex hull of any set of points and the upper envelope of any set of lines or planes have linear size triangulations. We conjecture that $\hat{\pi}_d(n,n) = \Omega(n^{\lfloor d/2\rfloor/(\lfloor d/2\rfloor+1)})$ for all d, but are unable to prove this when $d = 4$ or $d \ge 6$.

5. Polyhedral Partitioning Algorithms

The definitions and proofs in this section are almost exactly the same as those in Section 4 of [16]. The details are included here to keep the paper self-contained.

A *polyhedral partition graph* is a directed acyclic graph with one source, called the *root*, and several sinks, called *leaves*. Associated with each non-leaf node v is a set $\mathcal{R}_v$ of *query regions*, satisfying three conditions.

1. The cardinality of $\mathcal{R}_v$ is at most some constant $\Delta \ge 2$.
2. Every query region is a projective polyhedron with at most r faces, for some constant r.
3. The union of the regions in $\mathcal{R}_v$ is $\mathbb{R}^d$.

Typical values for r might be 2^{d+1} (every query region is a simplex) or 3^d+1 (every query region is a combinatorial cube). The query regions need not be disjoint. In addition, every non-leaf note v is either a *primal node* or a *dual node*, depending on whether its query regions $\mathcal{R}_v$ should be interpreted as a partition of primal or dual space. Each query region in $\mathcal{R}_v$ corresponds to an outgoing edge of v. Thus, the outdegree of the graph is at most Δ. There is no restriction on the indegree.

Given sets P of points and H of hyperplanes as input, a *polyhedral partitioning algorithm* for the halfspace emptiness problem constructs a polyhedral partition graph, which can depend arbitrarily on the input, and uses it to drive the following divide-and-conquer process. The algorithm starts at the root and proceeds through the graph in topological order. At every node except the root, points and hyperplanes are passed in along incoming edges from preceding nodes. For each node v, let $P_v \subseteq P$ denote the points and $H_v \subseteq H$ the hyperplanes that reach v; at the root, we have $P_{\text{root}} = P$ and $H_{\text{root}} = H$. At every non-leaf node v, the algorithm partitions the sets P_v and H_v into (not necessarily disjoint) subsets by the query regions $\mathcal{R}_v$ and sends these subsets out along outgoing edges to succeeding nodes. If v is a primal node, then for every query region $\Pi \in \mathcal{R}_v$, the points in P_v that are contained in Π and the hyperplanes in H_v whose lower halfspaces intersect Π traverse the corresponding outgoing edge. If v is a dual node, then for every $\Pi \in \mathcal{R}_v$, the points $p \in P_v$ whose dual hyperplanes p^* intersect or lie above Π and the hyperplanes $h \in H_v$ whose dual points h^* are contained in Π traverse the corresponding outgoing edge.[3] Note that a single point or hyperplane may enter or leave a node along several different edges.

For the purpose of proving lower bounds, the entire running time of the algorithm is given by charging unit time whenever a point or hyperplane traverses an edge. In particular, we do not charge for the construction of the partition graph or its query regions. As a result, partitioning algorithms have the full power of nondeterminism. In principle, any partitioning algorithm has "time" to compute the optimal partition graph for its input, and even very similar inputs might result in radically different partition graphs.

To solve the halfspace emptiness problem, the algorithm reports that all the points are above the hyperplanes if and only if no leaf in the partition graph is reached by both a point and a hyperplane. It is easy to see that if some point on or below a hyperplane, then there is at least one leaf in every partition graph that is reached by both the point and the hyperplane. Thus, for any set of points and hyperplanes, a partition graph in which no leaf is reached by both a point and a hyperplane provides a *proof* that every point is above every hyperplane.

[3] Alternately, we could let the points whose dual hyperplanes intersect Π and the hyperplane whose dual points intersect or lie below Π traverse the edge. Using this alternate formulation has no effect on our results. In fact, we can allow our partition graphs to have *four* types of non-leaf nodes — primal or dual; point/halfspace or ray/hyperplane — without changing our results, or even significantly altering their proofs.

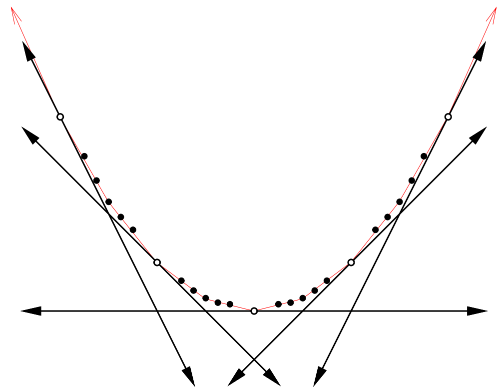

Figure 1. Worst case point-line configuration for halfspace emptiness; see Theorem 5.1. Tangent points are shown in white.

Theorem 5.1. *Any polyhedral partitioning algorithm that solves the halfspace emptiness problem in* $\mathbb{R}^d$*, for any* $d \geq 2$*, must take time* $\Omega(n \log m + m \log n)$ *in the worst case.*

Proof: It suffices to consider the following configuration, where n is a multiple of m. P consists of n points on the unit parabola $x_d = x_1^2/2$ in $\mathbb{R}^d$, and H consists of m hyperplanes tangent to the parabola and orthogonal to the (x_1, x_d) plane, placed so that n/m points lie between adjacent points of tangency. All the points in P are above all the hyperplanes in H. The dual points H^* also lie on the parabola $x_d = x_1^2/2$, and the dual hyperplanes P^* are also tangent to that parabola. See Figure 1.

For any point, we call the hyperplane whose tangent point is closest in the positive x_1-direction the point's *partner*. Every hyperplane is the partner of n/m points. A node v *splits* a point-hyperplane pair if both the point and the hyperplane reach v, and none of the outgoing edges of v is traversed by both the point and the hyperplane. A hyperplane h is *active at level* k if no node in the first k levels splits h from any of its partners.

Suppose v is a primal node. For each hyperplane h that v splits from one of its partner points p, mark some query polyhedron $\Pi \in \mathcal{R}_v$ that contains p but misses h. Since Π has at most r faces, the intersection of Π and the parabola consists of at most r arcs, so Π can be marked at most r times. Since there are at most Δ polyhedra in $\mathcal{R}_v$, at most $r\Delta$ hyperplanes become inactive at v. Similarly, if v is a dual node, then v splits at most $r\Delta$ points from their partners.

Thus, the number of hyperplanes that are inactive at level k is less than $r\Delta^{k+2}$. In particular, at level $\lfloor \log_\Delta(m/r) \rfloor - 3$, at least $m(1 - 1/\Delta)$ hyperplanes are still active. It follows that at least $n(1 - 1/\Delta)$ points each traverse at least $\lfloor \log_\Delta (m/r) \rfloor - 3$ edges. We conclude that the total running time of the algorithm is at least

$$n(1 - 1/\Delta)(\lfloor \log_\Delta (m/r) \rfloor - 3) = \Omega(n \log m).$$

Symmetric arguments establish a lower bound of $\Omega(m \log n)$ when $n < m$. □

The restriction to polyhedral partitioning algorithms is necessary for the lower bound to hold, since the problem can be solved in *linear* time in the generic partitioning algorithm model. The partition graph consists of a single primal node with two query regions: the convex hull of the points and its complement. If every point is above every hyperplane, then no hyperplane intersects the convex hull of the points.

This lower bound is tight, up to constant factors, in two and three dimensions.

Theorem 5.2. *Let $\mathcal{A}$ be an polyhedral partitioning algorithm that solves the halfspace emptiness problem, and let P be a set of points and H a set of hyperplanes, such that every point is above every hyperplane. Then $T_{\mathcal{A}}(P, H) = \Omega(\pi(P, H))$.*

Proof: Recall that the running time $T_{\mathcal{A}}(P, H)$ is defined in terms of the edges of the partition graph as follows.

$$T_{\mathcal{A}}(P, H) \triangleq \sum_{\text{edge } e} (\#\text{points traversing } e + \#\text{hyperplanes traversing } e)$$

We say that a point or hyperplane *misses* an edge from v to w if it reaches v but does not traverse the edge. (It might still reach w by traversing some other edge.) For every edge that a point or hyperplane traverses, there are at most $\Delta - 1$ edges that it misses.

$$\Delta \cdot T_A(P, H) \geq \sum_{\text{edge } e} (\#\text{points traversing } e + \#\text{points missing } e + \#\text{hyperplanes traversing } e + \#\text{hyperplanes missing } e)$$

Call any edge that leaves a primal node a primal edge, and any edge that leaves a dual node a dual edge.

$$\Delta \cdot T_A(P, H) \geq \sum_{\substack{\text{primal} \\ \text{edge } e}} (\#\text{points traversing } e + \#\text{hyperplanes missing } e) + \sum_{\substack{\text{dual} \\ \text{edge } e}} (\#\text{hyperplanes traversing } e + \#\text{points missing } e)$$

For each primal edge e, let P_e be the set of points that traverse e, and let H_e be the set of hyperplanes that miss e. The edge e is associated with a query polyhedron Π. Every point in P_e is contained in Π, and every hyperplane in H_e is disjoint from Π. Since Π has at most r faces, P_e and H_e are r-separable.

Similarly, for each dual edge e, let H_e be the hyperplanes that traverse it, and P_e the points that miss it. The associated query polyhedron Π separates the dual points H_e^* and the dual hyperplanes P_e^*. By the definition of dual polyhedra, Π^* separates P_e and H_e.

For every point $p \in P$ and hyperplane $h \in H$, there is node that splits them, since otherwise the algorithm would return the wrong answer, and thus some edge e such that $p \in P_e$ and $h \in H_e$. It follows that the collection of subset pairs $\{(P_e, H_e)\}$ is an r-polyhedral cover of P and H whose size is at least $\Delta \cdot T_{\mathcal{A}}(P, H)$ and, by definition, at most $\pi_r(P, H)$. □

We emphasize that every point must be above every hyperplane for this lower bound to hold. If some point lies below a hyperplane, then the trivial partitioning algorithm, whose partition graph consists of a single leaf, correctly "detects" the pair at no cost. This is consistent with the intuition that it is trivial to prove that some point lies below a hyperplane, but proving that every point lies above every hyperplane is more difficult.

Corollary 5.3. *The worst-case running time of any polyhedral partitioning algorithm that solves the halfspace emptiness problem in $\mathbb{R}^D$ is $\Omega(n \log m + n^{2/3} m^{2/3} + m \log n)$ for all $D \geq 5$ and*

$$\Omega\left(n \log m + \sum_{i=2}^{D} (n^{1 - \frac{2}{i(i+1)}} m^{\frac{2}{i+1}} + n^{\frac{2}{i+1}} m^{1 - \frac{2}{i(i+1)}}) + m \log n\right)$$

for all $D \geq d(d+3)/2$.

Proof: Theorems 5.1 and 5.2 together imply that the worst case running time is $\Omega(n \log m + \pi_d(n, m) + n \log m)$. The lower bounds then follow immediately from Theorem 4.6 and 4.7. □

Partitioning algorithms for the halfspace emptiness problem can (and do [12, 19]) apply a version of the "containment shortcut" described in [16]. If some query region lies entirely in a hyperplane's lower halfspace, then the hyperplane need not traverse the corresponding edge. Instead, if any point lies in that region, we immediately halt and report that some point is below a hyperplane. Although this shortcut decreases the running time of the algorithm, we easily verify that Theorem 5.2 still applies in the faster model.

Our techniques allow us to slightly improve earlier lower bounds for Hopcroft's problem in higher dimensions [16]: Given a set of points and hyperplanes, does any point lie on a hyperplane?

Theorem 5.4. *Let $\mathcal{A}$ be an polyhedral partitioning algorithm that solves Hopcroft's problem, and let P be a set of points and H a set of hyperplanes such that $I(P,H) = 0$. Then $T_{\mathcal{A}}(P,H) = \Omega(\pi(P,H))$.*

Combining this theorem with Theorem 4.5, we conclude:

Corollary 5.5. *The worst-case running time of any polyhedral partitioning algorithm that solves Hopcroft's problem in $\mathbb{R}^d$ is*

$$\Omega\left(n\log m + \sum_{i=2}^{d}\left(n^{1-\frac{2}{i(i+1)}}m^{\frac{2}{i+1}} + n^{\frac{2}{i+1}}m^{1-\frac{2}{i(i+1)}}\right) + m\log n\right).$$

This matches earlier lower bounds for the *counting* version of Hopcroft's problem.

6. Conclusions and Open Problems

We have proven a lower bound of $\Omega(n^{4/3})$ on the complexity of the halfspace emptiness problem in five or more dimensions. Our lower bounds apply to a broad class of geometric divide-and-conquer algorithms that recursively partition their input by divisions of space into constant-complexity polyhedra.

The most obvious open problem is to improve our results in dimensions other than five. The correct complexity in d dimensions is almost certainly $\Theta(n^{2-2/\lfloor d/2\rfloor})$, but we achieve this bound only when $d = 5$. In particular, the four dimensional case is wide open. It is not even known whether the four-dimensional halfspace emptiness problem is harder (or easier) than Hopcroft's problem in the plane [17].

The inner product doubling maps σ_d can be used to reduce Hopcroft's problem in $\mathbb{R}^d$ to halfspace emptiness in $\mathbb{R}^{d(d-3)/2}$ in linear time. Is there an efficient reduction from Hopcroft's problem to halfspace emptiness that only increases the dimension by a constant factor (preferably two)?

Our lower bounds are ultimately based on the construction of point-hyperplane configurations whose incidence graphs have several edges but no large complete bipartite subgraphs. Better such configurations would immediately lead to better lower bounds. Lower bounds in the Fredman/Yao semigroup arithmetic model have a similar basis. For example, Chazelle's lower bounds for offline simplex range searching [9] is based on a similar configuration of points and slabs. (See also [11].) Can we derive better polyhedral cover size bounds for points and hyperplanes from these configurations?

Another open problem is to prove tight lower bounds for *online* halfspace range query problems. Brönnimann, Chazelle, and Pach [5] have proven time-space tradeoffs for halfspace counting data structures in the Fredman/Yao semigroup model. Specifically, they prove that any data structure that uses space $n \le s \le n^d$ has worst-case query time

$$\Omega\left(\frac{(n/\log n)^{1-\frac{d-1}{d(d+1)}}}{s^{1/d}}\right).$$

Results of Matoušek [23] imply the upper bound $O((n/s^{1/d})\,\mathrm{polylog}\, n)$, which is almost certainly optimal (except possibly for the polylog factor), so the lower bounds have significant room for improvement. Chazelle and Rosenberg [11] have developed quasi-optimal tradeoffs for simplex reporting data structures in Tarjan's pointer machine model, but no lower bounds are known for *halfspace* reporting. No lower bounds are known for online halfspace emptiness queries in any model of computation. One possible approach, suggested by Pankaj Agarwal (personal communication), is to model range query data structures with partition graphs, and to prove tradeoffs between the total size of the graph (space) and the size of the subgraph induced by a query range (time).

A problem closely related to halfspace range searching is linear programming. The best known data structures of linear programming queries are based on data structures for halfspace emptiness [22] and halfspace reporting queries [6]. However, no nontrivial lower bounds are known for linear programming queries in any model of computation. One application of particular interest is deciding, given a set of points, whether every point is a vertex of the set's convex hull. Bounds for this problem closely match the best known bounds for halfspace emptiness [7], but the best known lower bound is $\Omega(n \log n)$. It seems unlikely that lower bounds can be derived for this problem in the partitioning algorithm model, since the extremity of a point depends on several other points arbitrarily far away.

Finally, extending our lower bounds into more traditional models of computation, such as algebraic decision trees or algebraic computation trees, is an important and extremely difficult open problem. A lower bound bigger than $\Omega(n \log m + m \log n)$ for *any* offline range searching problem in these models would be a major breakthrough.

References

[1] P. K. Agarwal, D. Eppstein, and J. Matoušek. Dynamic half-space reporting, geometric optimization, and minimum spanning trees. In *Proc. 33rd Annu. IEEE Sympos. Found. Comput. Sci.*, pages 80–89, 1992.

[2] P. K. Agarwal and J. Matoušek. Ray shooting and parametric search. *SIAM J. Comput.*, 22(4):794–806, 1993.

[3] A. Aggarwal, M. Hansen, and T. Leighton. Solving query-retrieval problems by compacting Voronoi diagrams. In *Proc. 22nd Annu. ACM Sympos. Theory Comput.*, pages 331–340, 1990.

[4] M. Ben-Or. Lower bounds for algebraic computation trees. In *Proc. 15th Annu. ACM Sympos. Theory Comput.*, pages 80–86, 1983.

[5] H. Brönnimann, B. Chazelle, and J. Pach. How hard is halfspace range searching. *Discrete Comput. Geom.*, 10:143–155, 1993.

[6] T. M. Chan. Fixed-dimensional linear programming queries made easy. In *Proc. 12th Annu. ACM Sympos. Comput. Geom.*, pages 284–290, 1996.

[7] T. M. Y. Chan. Output-sensitive results on convex hulls, extreme points, and related problems. In *Proc. 11th Annu. ACM Sympos. Comput. Geom.*, pages 10–19, 1995.

[8] B. Chazelle. Lower bounds on the complexity of polytope range searching. *J. Amer. Math. Soc.*, 2:637–666, 1989.

[9] B. Chazelle. Lower bounds for off-line range searching. In *Proc. 27th Annu. ACM Sympos. Theory Comput.*, pages 733–740, 1995.

[10] B. Chazelle, L. J. Guibas, and D. T. Lee. The power of geometric duality. *BIT*, 25:76–90, 1985.

[11] B. Chazelle and B. Rosenberg. Simplex range reporting on a pointer machine. *Comput. Geom. Theory Appl.*, 5:237–247, 1996.

[12] K. L. Clarkson. New applications of random sampling in computational geometry. *Discrete Comput. Geom.*, 2:195–222, 1987.

[13] K. L. Clarkson. A Las Vegas algorithm for linear programming when the dimension is small. In *Proc. 29th Annu. IEEE Sympos. Found. Comput. Sci.*, pages 452–456, 1988.

[14] D. P. Dobkin and D. G. Kirkpatrick. Determining the separation of preprocessed polyhedra – a unified approach. In *Proc. 17th Internat. Colloq. Automata Lang. Program.*, volume 443 of *Lecture Notes in Computer Science*, pages 400–413. Springer-Verlag, 1990.

[15] H. Edelsbrunner. *Algorithms in Combinatorial Geometry*, volume 10 of *EATCS Monographs on Theoretical Computer Science*. Springer-Verlag, Heidelberg, West Germany, 1987.

[16] J. Erickson. New lower bounds for Hopcroft's problem. In *Proc. 11th Annu. ACM Sympos. Comput. Geom.*, pages 127–137, 1995.

[17] J. Erickson. On the relative complexities of some geometric problems. In *Proc. 7th Canad. Conf. Comput. Geom.*, pages 85–90, 1995.

[18] J. E. Goodman and R. Pollack. Allowable sequences and order types in discrete and computational geometry. In J. Pach, editor, *New Trends in Discrete and Computational Geometry*, volume 10 of *Algorithms and Combinatorics*, pages 103–134. Springer-Verlag, 1993.

[19] J. Matoušek. Reporting points in halfspaces. *Comput. Geom. Theory Appl.*, 2(3):169–186, 1992.

[20] J. Matoušek and O. Schwarzkopf. On ray shooting in convex polytopes. *Discrete Comput. Geom.*, 10(2):215–232, 1993.

[21] J. Matoušek, M. Sharir, and E. Welzl. A subexponential bound for linear programming. In *Proc. 8th Annu. ACM Sympos. Comput. Geom.*, pages 1–8, 1992.

[22] J. Matoušek. Linear optimization queries. *J. Algorithms*, 14:432–448, 1993.

[23] J. Matoušek. Range searching with efficient hierarchical cuttings. *Discrete Comput. Geom.*, 10(2):157–182, 1993.

[24] N. Megiddo. Linear programming in linear time when the dimension is fixed. *J. ACM*, 31:114–127, 1984.

[25] R. Seidel. Small-dimensional linear programming and convex hulls made easy. *Discrete Comput. Geom.*, 6:423–434, 1991.

[26] J. M. Steele and A. C. Yao. Lower bounds for algebraic decision trees. *J. Algorithms*, 3:1–8, 1982.

[27] J. Stolfi. *Oriented Projective Geometry: A Framework for Geometric Computations.* Academic Press, New York, NY, 1991.

[28] G. M. Ziegler. *Lectures on Polytopes*, volume 152 of *Graduate Texts in Mathematics.* Springer-Verlag, 1994.

Appendix: Proof of Technical Lemma 3.1

There are two cases to consider — either the hyperplanes in H do not have a common intersection, or they intersect in a common flat. The proof of second case relies on the first.

Case 1 ($\bigcap H = \emptyset$):

Any polyhedron that separates P and H must be completely contained in a closed full-dimensional cell of the arrangement of H. It suffices to show, for each closed d-cell $\mathcal{C}$, that the set of n-point configurations contained in $\mathcal{C}$ and r-separable from H is topologically closed. Our approach is to show that this set is actually a compact semialgebraic set.

Fix a cell $\mathcal{C}$. Since the hyperplanes in H do not have a common intersection, both $\mathcal{C}$ and any polyhedra it contains must be polytopes. By choosing an appropriate hyperplane "at infinity" that misses $\mathcal{C}$, we can treat $\mathcal{C}$ and any polytopes it contains as *convex* polytopes in $\mathbb{R}^d$.

Let $A = \{a_1, a_2, \ldots, a_v\}$ and $B = \{b_1, b_2, \ldots, b_v\}$ be two indexed sets points in $\mathbb{R}^d$, for some integer v. We say that A *is simpler than* B, written $A \sqsubseteq B$, if for any subset of B contained in a facet of $\mathrm{conv}(B)$, the corresponding subset of A is contained in a facet of $\mathrm{conv}(A)$.[4] Equivalently, $A \sqsubseteq B$ if and only if for any subset of $d+1$ points in B, d of whose vertices lie on a facet of $\mathrm{conv}(B)$, the corresponding simplex in A either has the same orientation or is degenerate. Simpler point sets have less complex convex hulls — if $A \sqsubseteq B$, then $\mathrm{conv}(A)$ has no more vertices, facets, or faces than $\mathrm{conv}(B)$. If both $A \sqsubseteq B$ and $B \sqsubseteq A$, then the convex hulls of A and B are combinatorially equivalent.

If B is fixed, then the relation $A \sqsubseteq B$ can be encoded as the conjunction of at most $\mathrm{O}(v^{\lfloor d/2 \rfloor + 1})$ algebraic inequalities of the form

$$\begin{vmatrix} 1 & a_{i_0 1} & a_{i_0 2} & \cdots & a_{i_0 d} \\ 1 & a_{i_1 1} & a_{i_1 2} & \cdots & a_{i_1 d} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & a_{i_d 1} & a_{i_d 2} & \cdots & a_{i_d d} \end{vmatrix} \Diamond\, 0,$$

where $\Diamond$ is either $\geq$, $=$, or $\leq$. In every such inequality, the points $b_{i_1}, b_{i_2}, \ldots, b_{i_d}$ all lie on a facet of $\mathrm{conv}(B)$. For every d-tuple of points in B contained in a facet of $\mathrm{conv}(B)$, there are $v-d$ such inequalities, one for every other point. (If we replace the loose inequalities $\leq, \geq$ with strict inequalities $<, >$, the resulting expression encodes the combinatorial equivalence of $\mathrm{conv}(A)$ and $\mathrm{conv}(B)$.)

We can encode the statement "P is contained in $\mathcal{C}$ and is r-separable from H" as the following elementary formula:

$$\bigvee_{v=1}^{r} \quad \bigvee_{\substack{B \in (\mathbb{R}^d)^v \\ \mathrm{conv}(B) \text{ has at most } r \text{ faces}}} \left\{ \begin{array}{l} \exists\, a_1, a_2, \ldots, a_v \in \mathcal{C} : \\ \exists\, \lambda_1, \lambda_2, \ldots, \lambda_n \in [0,1]^v : \\ \qquad (A \sqsubseteq B) \;\wedge \\ \qquad \displaystyle\bigwedge_{i=1}^{n} \left(\sum_{j=1}^{v} a_j \lambda_{ij} = p_i \;\wedge\; \sum_{j=1}^{v} \lambda_{ij} = 1 \right) \end{array} \right\} \qquad (*)$$

Equivalently, in English:

> For some integer v, and for some set B of v points whose convex hull has at most r faces, there exists a set A of v points in $\mathcal{C}$, such that A is simpler than B (so $\mathrm{conv}(A)$ has at most r faces) and every point in P is a convex combination of points in A (with barycentric coordinates Λ).

Since there are only a finite number of combinatorial equivalence classes of convex polytopes with v vertices [18], the formula is actually finite, and therefore defines a semi-algebraic set. It remains only to show that this set is closed.

For any fixed v and B, the set of configurations $P \times A \times \Lambda \in (\mathbb{R}^d)^n \times \mathcal{C}^v \times ([0,1]^v)^n$ satisfying the subexpression

$$(A \sqsubseteq B) \;\wedge\; \bigwedge_{i=1}^{n} \left(\sum_{j=1}^{v} a_j \lambda_{ij} = p_i \;\wedge\; \sum_{j=1}^{v} \lambda_{ij} = 1 \right)$$

is the intersection of the closed convex polytope $\mathcal{C}^{n+v} \times [0,1]^{vn}$, vn hyperplanes, vn quadratic surfaces, and at most $(v-d)r$ closed algebraic halfspaces of degree d, and is therefore both closed and bounded. It follows that the set of point configurations P satisfying the subexpression of $(*)$ in braces is the projection of a compact set, and is therefore also compact. Finally, the set of configurations P satisfying the entire formula $(*)$ is the union of a finite number of compact sets, and therefore must be compact.

This completes the proof of Case 1.

Case 2 ($\bigcap H \neq \emptyset$):

The previous argument will not work in this case, because the cells in the arrangement of H are not simply connected, and thus are not polytopes.

Let P be an arbitrary set of n points in $\mathbb{RP}^d$, such that P and H are *not* r-separable. To prove the lemma, it suffices to show that there is an open neighborhood $\mathcal{U} \subseteq (\mathbb{RP})^d$ with $P \in \mathcal{U}$, such that for all $Q \in \mathcal{U}$, Q and H are not r-separable.

For any subset $X \subseteq \mathbb{RP}^d$ and any flat f, the *suspension of X by f*, denoted $\mathrm{susp}_f(X)$, is formed by replacing each point in X by the span of that point and f:

$$\mathrm{susp}_f(X) \triangleq \bigcup_{p \in X} \mathrm{span}(p \cup f).$$

The suspension of a subset of projective space roughly corresponds to an infinite cylinder over a subset of an affine space, at least when the apex of suspension is "at infinity". The *projection of X by f*, denoted $\mathrm{proj}_f(X)$, is the intersection of the suspension and the dual flat f^*:

$$\mathrm{proj}_f(X) \triangleq \mathrm{susp}_f(X) \cap f^*,$$

[4] Every set of points is simpler than itself. It would be more correct, but also more awkward, to say "A is at least as simple as B".

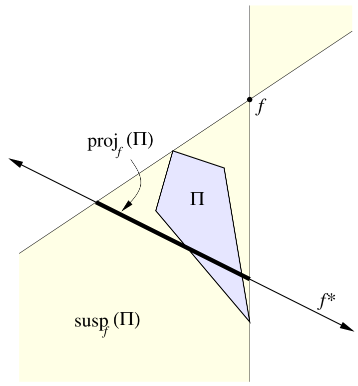

Figure 2. The suspension (double wedge) and projection (line segment) of a polygon by a point.

In particular, $\mathrm{susp}_f(X)$ is the set of all points in $\mathbb{RP}^d$ whose projection by f is in $\mathrm{proj}_f(X)$. The projection of a subset of projective space corresponds to the orthogonal projection of a subset of affine space onto a flat. See Figure 2.

Let $f = \bigcap H$, and let f^* be its dual flat. Without loss of generality, suppose the points $p_1, p_2, \ldots, p_m \in P$ are disjoint from f, and the points $p_{m+1}, \ldots, p_n \in P$ are contained in f. Denote these two subsets of P by $P \setminus f$ and $P \cap f$, respectively. Note that either subset may be empty.

If any polyhedron Π separates P and H, then its projection $\mathrm{proj}_f(\Pi)$ separates the projected points $\mathrm{proj}_f(P)$ and the lower dimensional hyperplanes $H \cap f^*$. Conversely, if any polyhedron $\Pi \subseteq f^*$ separates $\mathrm{proj}_f(P)$ and $H \cap f^*$ then the suspension $\mathrm{susp}_f(\Pi)$ separates P and H. Thus, P and H are r-separable if and only if $\mathrm{proj}_f(P)$ and $H \cap f^*$ are r-separable.

Since P and H are not r-separable, neither are $\mathrm{proj}_f(P)$ and $H \cap f^*$. The lower-dimensional hyperplanes $H \cap f^*$ do not have a common intersection. Thus, Case 1 implies that the set of configurations $P' \in (f^*)^m$ such that P' and $H \cap f^*$ are r-separable is closed. It follows that there is an open set $\mathcal{U}' \subseteq (f^*)^m$, with $\mathrm{proj}_f(P) \in \mathcal{U}'$, such that for all $Q' \in \mathcal{U}'$, Q' and $H \cap f^*$ are not r-separable.

Let $\mathcal{U}'' \subseteq (\mathbb{RP}^d)^m$ be the set of m-point configurations P'' such that $\mathrm{proj}_f(P'') \in \mathcal{U}'$. Clearly, $\mathcal{U}''$ is an open neighborhood of $P \setminus f$, and no configuration in $Q'' \in \mathcal{U}''$ is r-separable from H.

Finally, if Q'' and H are not r-separable, then no superset of Q'' is r-separable from H. Let $\mathcal{U} = \mathcal{U}'' \times (\mathbb{RP}^d)^{n-m}$. Then $\mathcal{U}$ is an open neighborhood of P. Since every configuration $Q \in \mathcal{U}$ has a subset Q'' that is not r-separable from H, no $Q \in \mathcal{U}$ is r-separable from H.

This completes the proof of Case 2, and thus the entire technical lemma. $\square$

The method we used to encode the condition "$\mathrm{conv}(A)$ has at most r faces" may seem somewhat convoluted. If we replace $A \sqsubseteq B$ with "$\mathrm{conv}(A)$ is combinatorially equivalent to $\mathrm{conv}(B)$", we get exactly the same semi-algebraic set, without needing to define the partial order $\sqsubseteq$. Unfortunately, testing whether two convex polytopes are combinatorially equivalent requires *strict* inequalities, whose corresponding semi-algebraic sets are *open*.